|  |  |  |  |
| --- | --- | --- | --- |
| Description : itu |  |  |  |

Intégration module Odoo

Et transaction mobile Client/marchand

par

Toavina Nirina RALAMBOSOA

Mémoire présenté  
en vue de l’obtention du grade de Licence

En Informatique

Option Développement d’application

Octobre 2017

Jury :

M. Rojo Rabenanahary, président  
 M. Tahina Razafinjoelina, examinateur  
 M. Manitra Andriamitondra, encadreur professionnel

© Toavina RALAMBOSOA, 2017

Table des matières

Liste des tableaux iii

Liste des figures iv

Glossaire v

Avant-propos 6

L’IT University 6

L’Institution/Entreprise d’accueil 7

Remerciements 8

Introduction 9

1 Présentation du projet 10

1.1 Objectifs du projet 10

1.2 Planning de réalisation 11

1.3 Architecture 12

1.3.1 Structure du dossier 12

1.3.2 Architecture flux - React-native 13

1.4 Technologies utilisées 14

1.4.1 Module Odoo de taux de change 14

1.4.2 Transaction mobile Client/marchand 15

2 Réalisation de l’application 20

2.1 Analyse et conception 20

2.1.1 Analyse de l'existant 20

2.1.2 Conception de l’application 22

2.2 Développement par fonctionnalité ou module 24

2.2.1 Module de taux de change Odoo 24

2.2.2 Application mobile de transaction Client/marchand 28

2.3 État d’Analyse et Statistiques 37

2.3.1 Volume de transaction Ariary.net 37

2.4 Problèmes rencontrés et solutions 38

3 Évaluation du projet et connaissances acquises 38

3.1 Bilan pour l'entreprise 38

3.2 Bilan personnel 38

3.3 Extension et évolution de l’application 39

Conclusion 40

Bibliographie 41

Annexe i

# Liste des tableaux

[Tableau 1 : Classement des bases de données ii](#_Toc497745428)

# Liste des figures

[Figure 1 : Diagramme de Gantt regroupé 11](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748840)

[Figure 2 : Structure dossier de travail 12](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748841)

[Figure 3 : architecture et relation entre les parties du flux 13](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748842)

[Figure 4 : Différence entre code barre 2D et 1D 15](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748843)

[Figure 5 : Code barre personnalisé pour Ariary.net 16](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748844)

[Figure 6 : Yahoo Finance –Taux de change du 12 Septembre 2017 20](#_Toc497748845)

[Figure 7 : Banque Centrale Malgache - Taux de Change du 12 Septembre 2017 21](#_Toc497748846)

[Figure 8 : Réception d'argent et solde 23](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748847)

[Figure 9 : Historique et détails 23](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748848)

[Figure 10 : liste des tables du serveur de test 24](#_Toc497748849)

[Figure 11 : liste des taux de change de la BCM 24](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748850)

[Figure 12 : un extrait de la structure du site du BCM (tableaux imbriqués) 25](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748851)

[Figure 13 : Diagramme de séquence extraction de données 26](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748852)

[Figure 14 : Ajout de nouvelle source Odoo 27](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748853)

[Figure 15 : Code barre sans montant 29](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748854)

[Figure 16 : Code barre montant 29](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748855)

[Figure 17 : notification en arrière-plan 31](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748856)

[Figure 18 : Badge de numérotation 31](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748857)

[Figure 19 : notification rapide 31](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748858)

[Figure 20 : génération de l'identifiant et sauvegarde dans le serveur 32](#_Toc497748859)

[Figure 21 : Notification depuis le serveur 32](#_Toc497748860)

[Figure 22 : Solde de compte et liste des menus 33](#_Toc497748861)

[Figure 23 : Paramètres pour le code d'autorisation 34](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748862)

[Figure 24 : Système Oauth2 - Type code d'autorisation 36](#_Toc497748863)

[Figure 25 : Liste des appareils connectés 36](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748864)

[Figure 26 : Volume quotidien des transactions 37](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748865)

[Figure 27 : Volume mensuel des transactions 37](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748866)

[Figure 28 : liste des modes de paiement dans WordPress i](file:///C:\Users\Toavina%20RALAMBOSOA\OneDrive\Documents\stage\Memoire\Guide_et_mod_le_m_moire_-_ITU_-_juin_2017\Guide%20et%20modèle%20mémoire%20-%20ITU\MémoireITU-v7.docx#_Toc497748867)

# Glossaire

**Progiciel**. Un logiciel professionnel standard aux multiples fonctions avec un ensemble de programme paramétrable. Il vise à unifier le système d’information d’une entreprise

**WebScraping**. C’est l’ensemble des méthodes pour l’extraction du contenu d’un site web via des lignes de code si on n’a pas accès à des API

**Framework**. c’est un ensemble d’outils regroupé en un pour constituer les fondations de logiciels informatiques ou applications web dans le but de faciliter le travail et d’augmenter la procuvtivité d’un informaticien

**Applications Hybrides**. Ce sont des applications mobiles développé à base d’HTML/CSS et JavaScript dont le but est de créer une seule application avec une seule code source pour des multiples plateformes.

**Standalone app**[[1]](#footnote-1) : c’est une application qui peut être utilisée seul sans modules et connaissances complémentaires

**Services web ou API**.

**Protocole**.

**Backend**

**Site e-commerce**

**Faux services :**

**MobileMoney :** les services monétaires par les opérateurs mobiles (OrangeMoney, MVola, AirteMoney)

# Avant-propos

Le présent mémoire présente les résultats du travail effectué lors de mon stage de fin d’études de Licence en Informatique de l’IT University, stage effectué chez NIVO SA durant 3 mois, de juin à septembre 2017.

Afin de poser clairement le contexte de ce mémoire, je vais présenter succincte d’une part de l’IT University et d’autre part de mon entreprise d’accueil.

## L’IT University

Fondée en 2011, l’IT University (ou ITU) est une université privée, spécialisée en informatique, formant les jeunes bacheliers scientifiques :

* En trois ans, pour l‘obtention d’une Licence, option Développement d’applications, Réseaux et Bases de Données ou Web et Design
* En cinq ans, pour l’obtention d’un Master MBDS en coopération avec l’Université de Nice Sophia Antipolis - France

Étant une formation professionnalisante, l’ITU a tissé des liens forts avec ses partenaires industriels, dont l’opérateur convergent TELMA, le GOTICOM et la plupart des entreprises et institutions du secteur des TIC[[2]](#footnote-2). Ces partenaires participent effectivement à la formation par la fourniture de connexion Internet à haut débit, l’envoi de conférenciers ou encore par l’accueil des étudiants en stage.  
D’autre part, le corps enseignant de l’ITU est constitué intégralement de spécialistes de très haut niveau et obligatoirement actifs professionnellement dans leurs domaines respectifs.  
Enfin, l’ITU fait partie du programme Microsoft Imagine en plus d’être un Oracle Gold Partner.

## L’Institution/Entreprise d’accueil

La société NIVO SA, si on se réfère aux documents internes, est une entreprise familiale fondée en 2000. Au début, son activité se basait sur l’importation de médicament et des produits de marques comme « Le petit Marseillais ». Dans son développement et l’envie d’avoir ses propres produits, il crée son propre usine durant l’année 2006. La société se spécialise ainsi dans la fabrication et la commercialisation de produit d’entretien sans mettre de côté son activité de base (l’importation). Tous ses produits ont une signature unique : « nickel ». Le siège se trouve à Analakely et son usine de production à Andranoro Ambohibao.

Depuis l’année dernière, avec l’arrivée du nouveau directeur Manitra ANDRIAMITONDRA, la société adopte une restructuration, une modernisation et surtout une informatisation de ses méthodes de travail que ce soit du côté de l’administration ou au sein de son usine. Dans cette idée de modernisation, la société s’intéresse à l’idée innovante apportée par Ariary.net, une solution de paiement qui vise à mettre en place un nouveau système monétaire dans le pays. Un contrat de partenariat entre les 2 entités fut alors signé. La société Nivo SA participe alors au lancement de ce système et c’est l’une des causes de notre présence dans la société.

## Remerciements

Durant ce stage, bon nombre de personne m’ont soutenu et je tiens à leur exprimer ma gratitude et mes remerciements.

Tout d’abord, je remercie le bon Dieu car c’est grâce à lui que j’ai eu le temps et la force pour accomplir ce stage, ainsi la réalisation de ce mémoire.

Ensuite, je suis fier d’exprimer ma gratitude envers ma famille qui est toujours là et qui place toujours leur confiance en moi.

Je tiens à exprimer particulièrement mes remerciements aux personnes ci-dessous :

* Docteur Olivier ROBINSON, Recteur de l’IT University
* Monsieur Manitra ANDRIAMITONDRA, Directeur de l’entreprise NIVO SA et mon encadreur professionnel
* Monsieur Rojo RABENANAHARY, mon encadreur pédagogique au sein de l’IT University
* Le personnel de NIVO SA qui m’a bien accueilli dans l’entreprise durant mon stage
* Le personnel d’Ariary.net qui ont travaillé avec moi sur la réalisation de l’application
* Les professeurs de l’IT University qui m’ont partagé leur connaissance et qui ont contribué à ma formation durant ces 3 ans.
* Le programme Sésame de l’Association PROMES qui est le pilier de mon parcours depuis 4 ans.

Je tiens aussi à remercier mes amis et mes proches qui m’ont soutenu de près ou de loin à la réalisation de ce mémoire.  
Enfin, je remercie également les membres du comité étudiant pour leur soutien.

# Introduction

L’essentiel du travail que j’ai réalisé s’étale sur 2 projets bien distinctes dont l’un relie les deux entités partenaires (Nivo SA et Ariary.net). On peut englober le tout sur le thème de :

« Intégration module Odoo et transaction mobile Client/marchand ».

Dans chaque entreprise moderne, l’utilisation d’un logiciel de gestion d’entreprise est devenue indispensable. Il permet de simplifier la gestion en elle-même, le suivi de l’ensemble des activités, depuis les stocks jusqu’aux ventes et même le personnel.

La société Nivo SA ne passe pas à côté et met en place Odoo pour gérer ses activités.

Cependant, C’est un outil généralisé pour avoir plus de cible et ne peut répondre à des demandes spécifiques à une entreprise ou à un pays. On peut appeler cela des « limites ». Des extensions sont disponibles pour y remédier mais la situation géographique et économique de notre pays ne favorise pas du tout la création de celle-ci.

Faisant partie de ces limites, le taux de change dans le système présente des différences par rapport à la réalité (référence la Banque Centrale Malgache) mais avec l’amélioration fourni, Odoo peut désormais « mettre à jour automatiquement et sans erreurs ses taux de change ».

Ariary.net, de son côté, a déjà son système de paiement bien en place, opérationnel et disponible depuis son site web[[3]](#footnote-3). Tous les offres telles que les achats et les transactions y sont accessibles.

Le problème réside dans le fait qu’on utilise plutôt notre smartphone qu’un ordinateur dans notre quotidien et on privilégie les applications aux navigateurs. Un fait qui rend le système très peu accessible pour l’ensemble de la population. Avec l’application ariary.net, il sera possible très prochainement :

* D’avoir accès à son compte Ariary.net depuis son smartphone
* De faire des transactions simple, rapide et sécurisée
* De faire des achats en un clic
* D’avoir des notifications à chaque transaction effectuée

Ce présent mémoire sera résumé en 3 grandes parties à savoir :

* La présentation du projet
* La réalisation
* Évaluations et connaissances acquises

# Présentation du projet

## Objectifs du projet

Dans la première partie du stage, l’objectif du projet est d’extraire automatiquement les données sur le taux de change depuis le site[[4]](#footnote-4) de la Banque Centrale et de mettre à jour à partir de ces données celles dans Odoo.

Notons qu’une extension sous le nom de « currency\_rate\_update » fait déjà la même chose mais les sources de données disponibles sont :

* Yahoo finance
* Bank Of Canada
* Bank Of Mexico

De ce fait, pour la mise à jour des données dans Odoo, nous avons choisi de créer une extension basée et dépendante de l’existante pour suivre les normes placées et d’épargner le temps perdu à refaire ce qui a été déjà fait.

L’application mobile, quant à elle, a pour objectif de vulgariser les services proposés par Ariary.net, de créer un système de transaction et d’échange le plus simple, rapide et sécurisée possible. Tout cela, en utilisant l’objet le plus attaché à l’homme actuel, son « Smartphone » sans faire d’exception sur le type (IOS et Android).

## Planning de réalisation

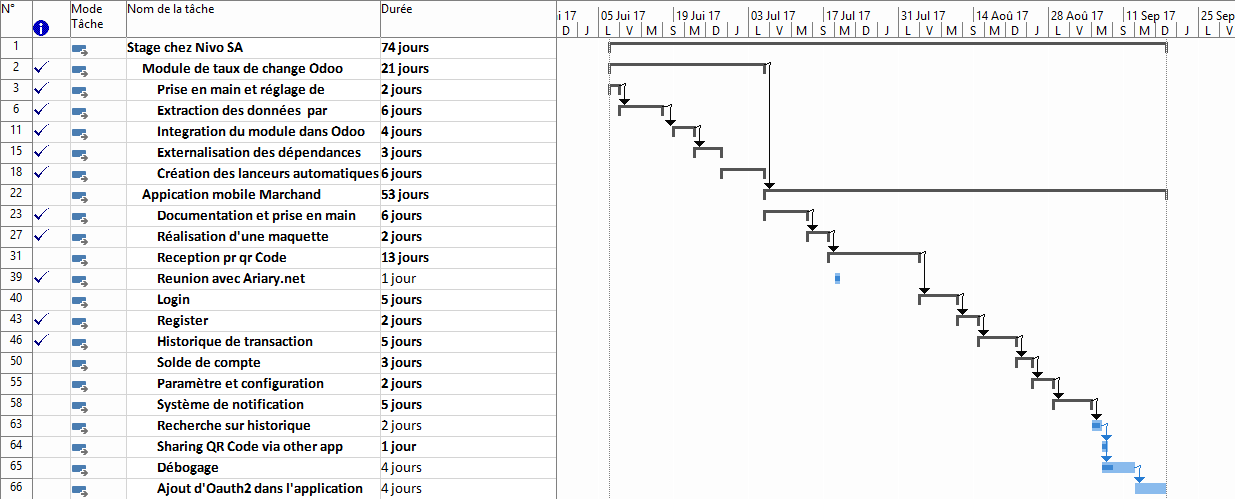
Le planning de réalisation des applications se présente comme suit (l’affichage est regroupée par étape de travail et fonctionnalités) :

Figure 1 : Diagramme de Gantt regroupé

## Architecture

### Structure du dossier

Dans les tous les projets que l’on a fait durant ce stage, on nous a appris à travailler avec une structure de dossier bien défini. Elle est définie en 4 parties bien distinctes :

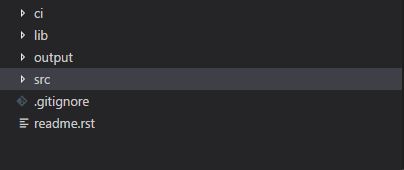
* Src : c’est le dossier contenant les codes sources propres à l’application.
* Lib : dossier contenant des librairies utilisées
* Output : Dossier de sortie d’application prête à l’emploi
* Ci/ : le dossier qui contient des « scripts bash » permettant de faire des lancements automatiques.
  + Build.sh : script permettant de faire automatiquement des actions suivantes :
    - Téléchargement des dépendances définies dans le packager et les prérequis
    - Copie de ceux qui se trouve dans le dossier « /lib » vers le dossier « src/ »
    - Compilation de l’ensemble pour créer un module ou application prête à l’utilisation dans le dossier « /output »
  + Test.sh : c’est un script pour le lancement des tests unitaires pour le projet en question (avec téléchargement des dépendances si besoins)

Figure 2 : Structure dossier de travail

### Architecture flux - React-native

Le framework que nous avons choisi, React-native est une dérivée pour mobile de React.js. Ces 2 Framework sont basés sur un architecture sous le nom de « flux ». Par rapport aux architectures MVC, le flux comporte 4 parties simples :

**Les actions** : l’ensemble de tout ce qui proviennent de l’utilisateur ou du serveur (interaction),

**Le dispatcher** : Il agit comme un intermédiaire sur le flux d’action, il reçoit les action et renvoi au bon destinataire

**Les stores** : ce sont les équivalents des classes et model dans d’autres architectures

**Les views :** C’est le responsable des rendus et sources des actions.

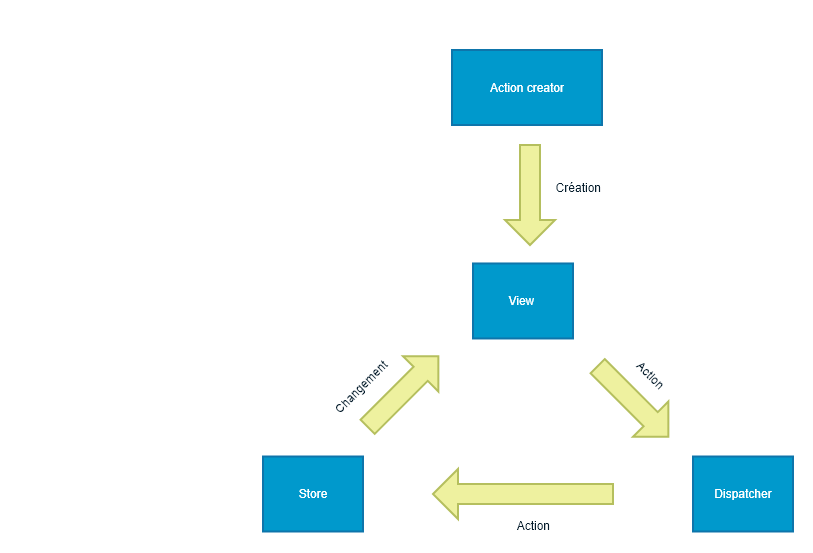
Sa particularité réside dans le fait que la relation entre les différentes parties est unidirectionnelle, les actions ne vont qu’à un sens et retentit dans une boucle. L’ensemble démarre grâce à une « Action creator ».

Figure 3 : architecture et relation entre les parties du flux

## Technologies utilisées

### Module Odoo de taux de change

**Le choix d’ERP : Odoo 9.0**

Sur le choix de progiciel au sein de la société, on a opté pour Odoo version 9.0. En effet, il est le leader des progiciels « open Source », son système à évolution rapide décrit sa force avec une communauté de développeurs large permettant d’avoir plus de 8000 modules disponibles et 500 partenaires certifiés[[5]](#footnote-5). Avec un design très attractif et 100% Web, il permet une utilisation et une compréhension facile. Le fait d’être interopérable avec une multitude de services tiers (Les logiciels Microsoft, Google, …) avec une personnalisation presque illimitée renforcent son efficacité. Odoo est actuellement à sa 10ème version mais comme c’est assez récent, son prédécesseur présente encore plus de module disponible qu’elle et il arrive à combler les majeures parties des besoins de la société.

**Docker**

Docker est un outil qui permet d’empaqueter une application et toutes ses dépendances dans un conteneur isolé[[6]](#footnote-6). Au lieu d’installer directement Odoo sur l’ordinateur, on installe l’ensemble dans un conteneur (paquet) de Docker. Ceci permet une portabilité de l’application d’un serveur à une autre sans avoir à tout reconfigurer. Il permet aussi de bien définir la version des dépendances à utiliser sans toucher celui installé dans l’ordinateur.

**Module Currency\_\_rate\_update**

Ce module est une extension libre et openSource d’Odoo permettant de mettre à jour automatiquement le taux de change dans Odoo via des sources de données dans le monde. Nous avons choisi d’utiliser celle-ci au lieu d’un autre pour les raisons que

* L’un des références en taux de change dans le monde (Yahoo Finance) est disponible dans le module comme source de données
* Des sources de données comme … utilisent aussi le webscraping pour extraire ses données.
* L’ajout de nouvelle source de données est très rapide grâce à l’existence d’une classe d’implémentation.

**Langage de programmation Python**

Au niveau de développement dans Odoo, ils ont adopté Python 2.7, ce langage est déjà stable par sa maturité et il occupe la 5ème place (une place non négligeable) dans le classement de TIOBE de 2017. Il doit une partie de sa popularité par la clarté de ses lignes de code et l’installation simple rapide des librairies et modules avec « pip ».

**BeautifulSoup**

Étant donné que le site de la Banque centrale de Madagascar (BCM) ne présente aucun API pour accéder à ses données, nous avons opté pour l’utilisation du « WebScraping » pour extraire les données. Par rapport aux autres librairies d’extraction de données (Sélénium et lxml.html), il est le plus adapté à notre besoin qu’est l’extraction automatique de données depuis la page d’accueil de la BCM (page HTML). BeautifulSoup se base sur le système de hiérarchie des données pour faire des recherches et on peut faire une recherche à partir de mot clé (appropriée à des sites d’ancienne version).

### Transaction mobile Client/marchand

**Le QR Code – Code barre 2D**

Une code-barres est manière de présenter sous format numérique des données et des informations qu’on peut voir comme un symbole et qui est lu facilement par des appareils dites lecteurs de code-barres. Ces appareils peuvent être des appareils spécialisés ou simplement des smartphones.

Figure 4 : Différence entre code barre 2D et 1D

Les codes-barres 2D sont des formes avancées offrants plus de possibilité sur l’utilisation des codes-barres. Si le simple code-barres à une Dimension ne contient de l’information que dans l’horizontal et que sa taille varie selon celle des données qu’elles contiennent, les 2 dimensions permettent d’avoir des données sur les 2 axes avec une taille non dépendante de ses données.

Comme notre projet consiste à faire une échange rapide et simple, nous avons opté pour le choix des codes-barres à 2 Dimensions comme solution pour les transactions et plus précisément le QR Code « Quick Response Code », le code barre à 2 Dimensions le plus utilisé dans le monde. Ce choix du QR Code a été renforcé pour les raisons suivantes :

* Il est très facilement et rapidement lu par les smartphones. Cette lecture se fait par l’utilisation d’application dédiée, de l’appareil photo et le tour est joué.
* Sa capacité de stockage de données et largement grande et peuvent contenir jusqu’à 7089 caractères numériques et/ou 4296 caractères alphanumériques.
* Comme le taille du QR Code ne dépend pas de celle de ses données et que c’est en 2 dimensions, sa forme tient parfaitement sur un écran de smartphone.
* Sa forme et l’ajout de personnalisation est disponible, permettant ainsi de faire un QR code bien propre à Ariary.net.



Figure 5 : Code barre personnalisé pour Ariary.net

**Framework React-native**

React-native est un Framework JavaScript hybride[[7]](#footnote-7) développé par Facebook pour le développement d’application mobile. Nous n’avons pas hésité à faire ce choix vu les possibilités qu’il offre. En effet, react-native est en premier lieu un framework Open Source, et Hybride. De son slogan « Learn once, write everywhere », il permet de créer une application compatible à tous les systèmes mobiles existants (Android, IOS, Windows Phone, …) avec un seul code source écrit avec JavaScript. Le framework se démarque aussi des autres hybrides (Cordova, Ionic, …) du fait qu’il utilise les composants natives (propres aux systèmes) au lieu d’utiliser des « Webview » pour le rendu de l’application, il offre ainsi une performance bien au-dessus de ses concurrents.

Étant donné qu’il est en JavaScript et Open Source, une large communauté et des libraires externes (propre à react-native et les librairies JavaScripts[[8]](#footnote-8)) sont disponibles avec une facilité d’installation et d’utilisation.

**Expo Project**

C’est un ensemble d’outils, de services et de multiples librairies que l’on peut utiliser avec react-native. Il simplifie grandement le code source avec un développement et débogage en temps réel avec l’application Expo installé sur le mobile de test.

Il offre aussi la possibilité d’un système de notification basée sur l’identification du mobile et l’application. Une fonctionnalité indispensable pour notre application.

En utilisant l’Expo XDE[[9]](#footnote-9) (outils de développement) ou simplement la ligne de commande et la librairie expo, on peut générer une application portable (Standalone App) pour un plateforme choisi. Un manière rapide et efficace.

**Le protocole Oauth 2**

L’Oauth2 est une deuxième version du framework de sécurité Oauth. Son objectif est de permettre à une autre entité (application) d’avoir accès aux données et à l’API du serveur de façon sécurisée. En d’autre termes, c’est un protocole d’accès à des services web. Cette version en question est actuellement la norme utilisée pour avoir accès aux données des utilisateurs des grands conglomérats du monde de l’informatique (Facebook et Google…). L’objectif de ce protocole est de permettre à un client de parler aux serveurs seulement quand il a un clé d’accès valide du nom Token qui lui a été généré lors de sa connexion. Ce Token est une chaîne de caractère qui sera la seule porte pour avoir des données. Et on a 2 types de Token :

* Access Token : une clé permettant d’avoir des données mais d’une durée de validation limitée. Il peut être utilisé plusieurs fois.
* Refresh Token : De son nom, il permet d’avoir un nouvel Access Token quand celui que vous utilisé n’est plus valide. On ne peut l’utiliser qu’une seule fois.

Son utilisation permet ainsi de donner des informations et des accès qui, selon de type d’autorisation légué, pourront être utilisé par de différentes entités. On peut créer son propre type d’autorisation mais les bases sont :

**Autorisation Grant Code :** On obtient avec ce type d’autorisation un « Access Token » et un « Refresh Token ». Elle est surtout destinée aux clients dits confidentiels, c’est-à-dire, un client qui demande est un serveur.

**Implicit Grant** **:** Ce type ne permet que d’avoir un « Access Token » après que l’utilisateur se soit connecté sur une redirection. Elle est utilisée dans les applications mobiles tierces et les applications JavaScripts (dans des navigateurs)

**Credentials Grant :** Celle-ci n’est utilisé que lorsque le client et le serveur sont de la même entité, on parle ici d’un produit venant de la même entreprise. Il aura accès à un « Access Token » et un « Refresh Token ».

**Clients Credentials Grant :** Elle permet d’avoir un accès d’usage bien limité et bien défini à un client sur les données. On aura alors un « Access Token » en retour.

**Back-end – PHP**

Pour la création de notre serveur de test devenu une passerelle avec Ariary.net, nous avons choisi d’utiliser le langage PHP. C’est un langage de script généraliste et OpenSource conçu pour le développement web et la création de serveur. Ce choix est dû au fait que malgré sa simplicité il offre une diversité de fonctionnalités avancées. Il est aussi rapide et gratuit. Au niveau de portabilité, il marche très bien sur Windows, Linux ainsi que Mac sans aucune perte de performances.

Pour la génération et la gestion des notifications avec Expo, nous avons opté pour PHP par le simple fait que c’est un langage Orienté objet, dynamique. Mais aussi, il est bien adapté pour les webservices RESTful.

**SGBD - MySQL**

PostgreSQL et MySQL sont des systèmes de gestion de base de donnée (SGBD) gratuit et libre. Notre but ici est de créer un serveur passerelle entre le vrai serveur d’Ariary.net et l’application mobile. Son but est de mémoriser des données propres à l’application sans devoir faire des modifications au niveau de la base actuelle d’Ariary. Nous avons choisi MySQL par le fait qu’il est surtout destiné pour des petits et moyens volume de données.

# Réalisation de l’application

## Analyse et conception

### Analyse de l'existant

#### Insertion manuelle

Pour mettre à jour les données, une personne de l’entreprise est chargé chaque début de semaine pour le faire manuellement en allant sur le site web et modifier celui dans Odoo. Une tâche qui prenait environs 10mn et dont les données ne prennent pas alors en compte les fluctuations du reste de la semaine.

#### Extension « currency\_rate\_update »

C’est une extension, dans le Store d’Odoo, qui gère la mise à jour automatique des taux de change. Elle a de multiples sources de données dont la plus connue est « Yahoo finance ». Comme cette source supporte notre monnaie le MGA on peut faire appel à ses services pour les mises à jour des taux de change. Mais par rapport aux au sein de la banque centrale malgache[[10]](#footnote-10), on aperçoit des différences non négligeables qui pourrait amener à des erreurs de calcul.

A titre d’exemple, voici les taux de change du 12 Septembre 2017 selon Yahoo Finance et la Banque Centrale Malgache :

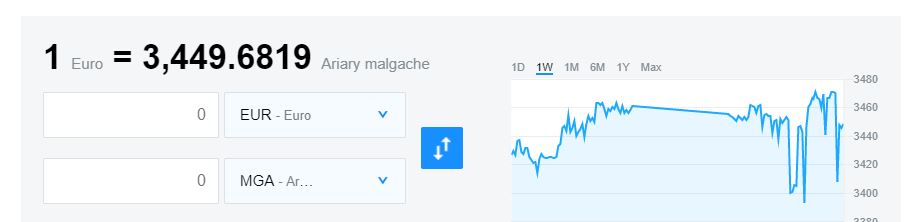


Figure 6 : Yahoo Finance –Taux de change du 12 Septembre 2017



Figure 7 : Banque Centrale Malgache - Taux de Change du 12 Septembre 2017

#### La Banque Centrale Malgache

Le site de la Banque Centrale Malgache est actuellement la référence en terme de taux de change à Madagascar. Malheureusement, il ne comporte aucun API jusqu’à maintenant pour l’accès aux données. De plus, la structure de leur site web ne suit pas la norme actuelle mais repose encore sur l’ancienne version (système de tableau imbriqué), ce qui ne facilite pas la lecture automatique des données (WebScraping).

#### Forme d’échange à Madagascar

Pour le cas de notre pays, on peut dire que la majorité des transactions se fait toujours avec de l’argent liquide (espèce). Un procédé qui possède de multiples inconvénients surtout l’insécurité et l’arrondissement des prix.

D’autres formes d’échange sont aussi disponible dans notre pays tels que les virements bancaires ou encore l’utilisation des Mobile Money. Mais leurs inconvénients résident soit dans ses procédures qui sont longues et avec beaucoup de formalité soit sur un fonctionnement compliqué et qui peut durer plusieurs minutes (trop de temps perdu si on veut juste acheter rapidement du jus ou autres produits de consommation dans une épicerie).

C’est pour ces raisons que nous avons mises en place notre système qui assure l’échange et payement en un temps record sans fléchir en terme de fiabilité et de sécurité.

#### Accès et fonctionnement d’Ariary.net

Jusqu’à maintenant, l’accès au plateforme d’Ariary.net ne peut se faire qu’à partir du web et d’un navigateur. En plus, pour avoir accès à la solution de paiement, l’inscription est devenue obligatoire et requiert beaucoup trop de champs. Bon nombre d’utilisateur font alors marche arrière à cause de cette complication. Le système n’a actuellement que très peu d’utilisateur et le taux d’inscription et très faible.

### Conception de l’application

Pour le module de taux de change, comme nous l’avons basé sur un autre, il n’y avait pas la nécessité de créer de nouvelle affichage mais seulement de créer une nouvelle source de données. Sans compter alors les questions d’affichage, la phase de conception consistait surtout à bien définir les étapes (et les fonctions qui vont avec) depuis l’extraction vers l’insertion des données. De ce fait, on a eu besoin de 2 classes de base pour la réalisation du module

* Utils.py : contenant les fonctions pour le WebScraping
* Update\_service\_BCM.py : pour l’ajout de nouvelle source dans le module de dépendance.

Pour l’autre partie du stage, rappelons tout d’abord qu’avant la création de cette application mobile, le système Ariary.net était déjà opérationnel. Il avait déjà ses propres services et ses propres sécurités mais ne suivait pas encore les normes du protocole Oauth2.

En prenant cela en compte, nous avons procédé à la conception de l’application de la manière suivante :

**Recherche d’application similaire et maquette** : En se souvenant d’une phrase que l’on nous a dit en deuxième année « On ne réinvente pas la roue » et du fait que déjà bon nombre de personne ont fait la même chose que nous auparavant, nous avons fait des recherches pour trouver des applications avec des fonctionnalités de base similaires et qui ont les mêmes ambitions que nous. Et c’est ainsi que l’on a trouvé 2 applications de référence :

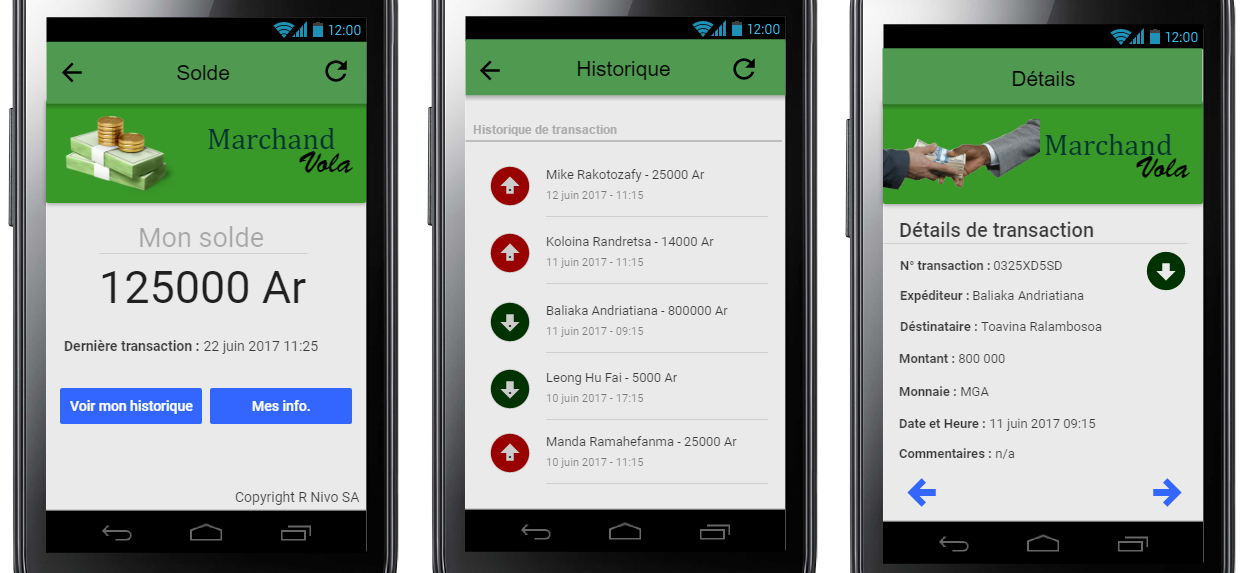
* **BlockChain** : application mobile pour Bitcoin basée aussi sur le QR Code (très similaire à Ariary.net)
* **Paypal** : l’une des solutions de paiement de plus utilisé dans le monde.

Nos inspirations se sont donc basées sur ces 2 applications

**Application mobile Client/marchand**

Pour la conception de l’application mobile, nous avons commencé par faire la liste des fonctionnalités de base et des écrans d’affichage qui vont avec. Puis nous avons mis en place les scénarios d’utilisation que l’on pourrait faire avec l’application pour mettre en place les liens entre les écrans d’affichage. Et c’est après que des maquettes de base ont été faites pour avoir un aperçu du produit.

Figure 8 : Réception d'argent et solde



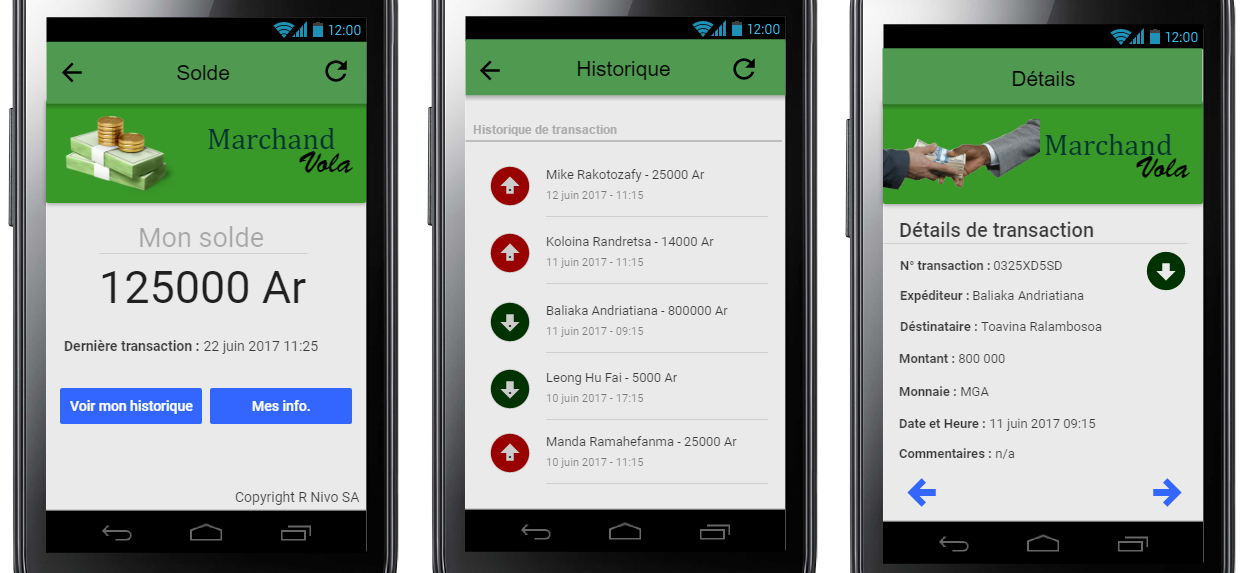


Figure 9 : Historique et détails

**Création d’un serveur de test :** Pour le serveur de test, nous avons en sorte de simplifier au maximum et de n’inclure que les informations indispensables. Ainsi, la base de donnée ne comporte que 2 tables : l’une pour les transactions et l’autre pour l’identification du compte et le système de notification

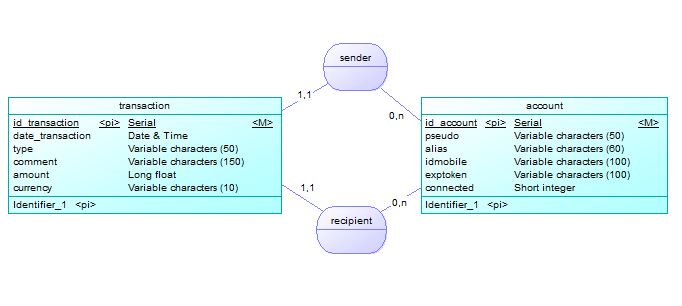


Figure 10 : liste des tables du serveur de test

## Développement par fonctionnalité ou module

### Module de taux de change Odoo

#### Réalisation d’un module externe de récupération de données

Pour ne pas s’embêter avec les problèmes liés à Odoo et pour simplifier la création du système d’extraction de données, nous avons décidé de créer en premier temps un module qui n’est pas liée à celui-ci mais un simple projet python permettant de faire le travail nécessaire. Les données en question sont la liste des taux de change de l’Ariary avec les autres unités monétaires.

Figure 11 : liste des taux de change de la BCM

Et dans ce genre de travail, l’une des façons pour avoir les données nécessaires est l’utilisation d’un API, une fonctionnalité non disponible ou bien non disponible au grand publique sur notre source de données, la Banque Centrale Malgache.

L’autre méthode est de faire du « WebScraping », il consiste à créer des fonctions capables de simuler des actions sur le navigateur et/ou de parcourir les lignes de codes de chaque page web pour y extraire les données voulues.

Avec python, il existe 3 principaux librairies qu’on a trouvé et qu’on a essayé d’utiliser pour la réalisation d’un webScraping :

* Sélénium (nécessite l’ouverture physique d’un navigateur)
* Lxml (spécialisé dans la lecture des fichiers XML)
* BeautifulSoup (notre choix – recherche basée sur la hiérarchie de données)

Notons que le Site de la Banque Centrale comporte les spécificités suivantes :

* Un très vieux site dont la forme se base sur des tableaux imbriqués
* Une utilisation presque invisible des identifiants (id) et classe permettant de rechercher les données par spécification.

Figure 12 : un extrait de la structure du site du BCM (tableaux imbriqués)

Le mieux adapté « BeautifulSoup » permettait de faire des recherches par les manières suivantes :

* Utilisation de la hiérarchie (relation entre un balise père et fils)
* Recherche par mots clés
* Recherche par identifiant ou classe

Et le module fonctionne comme suit pour trouver les données :

Il extrait d’abord l’ensemble de la page de donnée et transforme les données sous forme d’arbre.

Il recherche ensuite l’une des données par mot clés. (Nous avons fait en sorte de choisir un mot clé qui a le minimum de probabilité d’avoir des duplications sur les autres parties de la page). On a recherché dans l’arbre ce qui comportait le mot « XDR » : c’est l’un des unités monétaires et un mot que l’on pense revenir très rarement ou même impossible sur une autre partie d’un site en français.

Après cela, par le biais de la hiérarchie, on remonte l’échelon pour avoir la liste complète des taux de change.

Et enfin on extrait de chaque branche les données pour en faire un tableau.

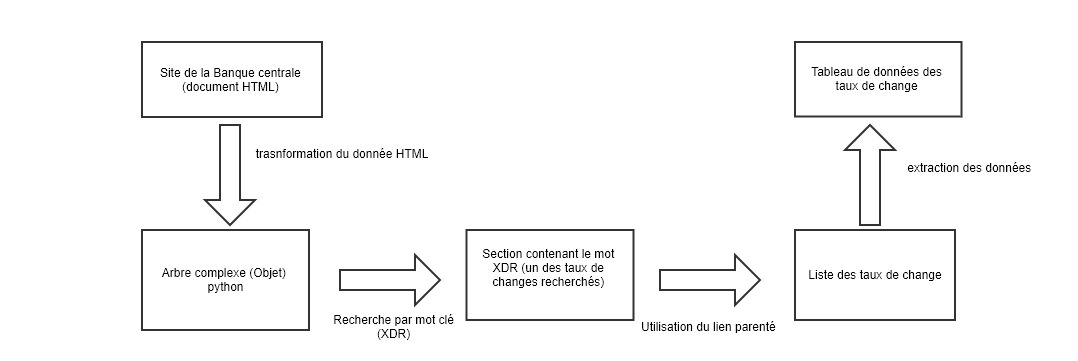


Figure 13 : Diagramme de séquence extraction de données

#### Ajout de nouvelle source à « currency\_rate\_update »

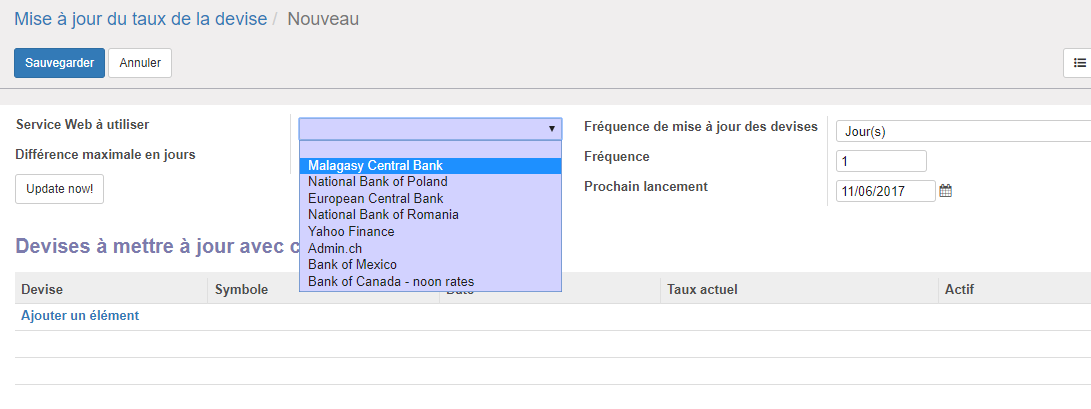
Après l’extraction des données sous format d’un tableau, l’étape suivante consiste à ajouter une nouvelle source dans le module dépendance. En implémentant la classe « CurrencyGetterInterface » dans notre propre module sous le nom de « Update\_service\_BCM », Odoo le reconnaîtra automatiquement et l’insèrera dans la liste des sources de données.

Figure 14 : Ajout de nouvelle source Odoo

#### Externalisation des dépendances et lanceurs automatiques

Pour suivre l’architecture de dossier que nous avons mis en place, nous avons fait en sorte d’externaliser toutes les dépendances utilisées (BeautifulSoup & Currecy\_rate\_update) et de ne mettre que les codes propres à notre module dans le dossier src/. La librairie BeautifulSoup a été déplacée dans lib tandis que currency rate\_update est téléchargé par « cURL [[11]](#footnote-11)».

### Application mobile de transaction Client/marchand

#### Réception d’argent par QR Code

Cette partie du projet est l’un des plus important de cette application car les transactions et les échanges en dépendent complètement.

Elle consiste entre autre à générer et d’afficher un QR code contenant tous les informations permettant de faire les transactions. Ces informations sont :

* L’identifiant du destinataire (ex : AA012)
* Son nom d’utilisateur (ex : Toavina)
* Le montant de la transaction (facultatif)
* L’unité monétaire
* Le type d’échange (ex : trans 🡪 pour désigner un échange de type transaction)

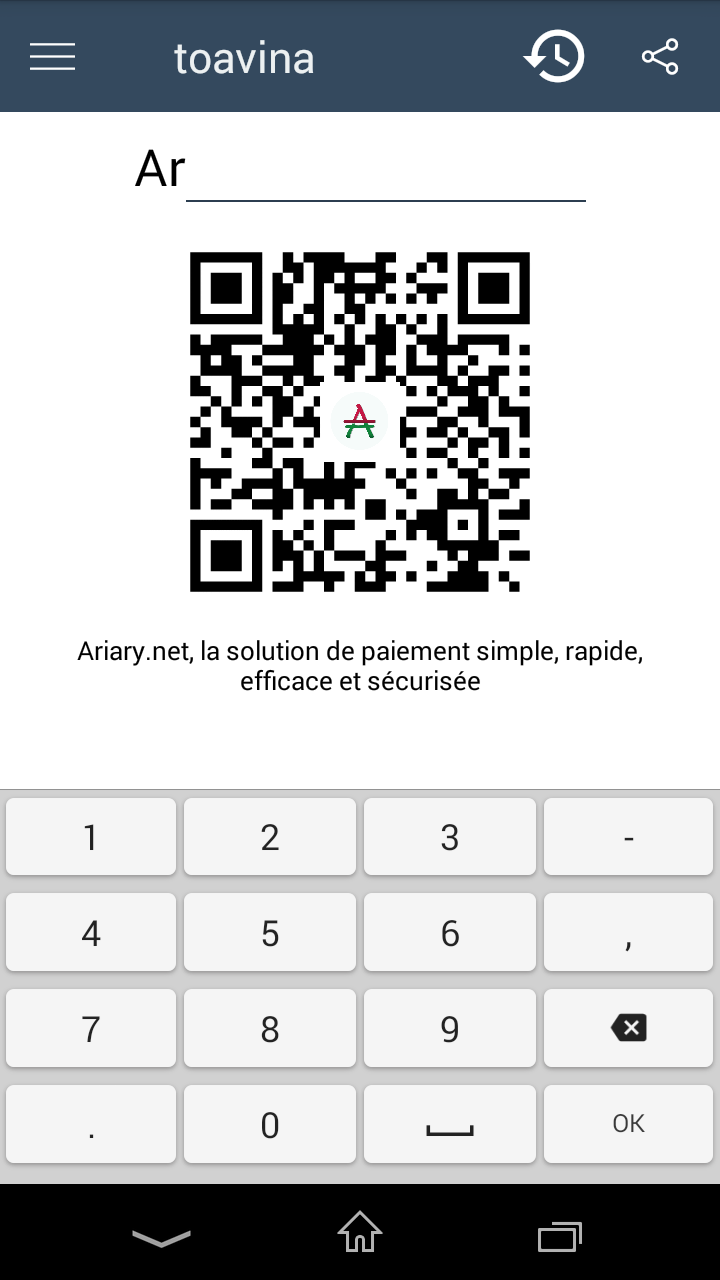
Pour faciliter l’utilisation de l’application, nous avons mis un champ permettant de définir le montant du transfert, il est initialement à zéro (0) et un changement en temps réel du QR Code est visible à l’œil nu quand on ajoute ou modifie le montant.

Pour l’utilisation de cette fonctionnalité, nous avons fait en sorte de faire une interface la plus facile à utiliser. Dès l’arrivée sur la page, l’utilisateur peut déjà recevoir de l’argent mais peut aussi spécifier le montant qu’il veut recevoir. Si le montant n’est pas défini, ce sera alors celui qui envoi qui le fera.

Pour faire la transaction, un autre utilisateur de l’application prend juste le QR Code en photo et on n’a plus qu’attendre. On aura alors une notification à la fin pour confirmer que la transaction a bien aboutit.

Figure 15 : Code barre sans montant

Figure 16 : Code barre montant



#### Système de notification – Fiabilité et Sécurité

Une notification est un signal ou une alerte permettant de dire ou de rappeler à un utilisateur d’une action qui vient de se produire, d’un changement ou bien le début ou l’aboutissement d’une activité.

Dans notre application, comme la principale action est la transaction monétaire, l’activité se base surtout sur la fiabilité et la sécurité du système. Dans cette question de fiabilité, on vise surtout à rassurer l’utilisateur sur l’aboutissement de la transaction qu’il a effectué. Pour faire cela, nous avons créé un système permettant d’alerter « toutes les mobiles connectés » au compte de l’utilisateur pour lui confirmer le succès et le type de transaction qui s’effectué depuis son compte.

Pour la bonne pratique, chaque mobile lié aux comptes reçoit la notification même si l’application Ariary.net n’est pas en cours d’exécution. Et pour ne pas faire les choses à moitié, nous avons mis 3 types de notification pour donner plus d’assurance à l’utilisateur :

**Notification en arrière-plan :** Ce type de notification marche même si l’application n’est pas en premier plan. Lors de l’aboutissement d’une transaction, une notification du genre « Vous avez envoyé 2000 Ar à Toavina » ou du style « Vous avez reçu 2000Ar de Hasina » apparait dans la zone de notification du téléphone. Elle est surtout utile pour vous avertir d’une transaction à votre insu.

**Notification rapide :** Celle-ci apparaîtra sur votre écran quand vous êtes dans l’application pour vous dire rapidement qu’un transfert vient d’être effectué. Elle apparaitra seulement pour quelques secondes pour ne pas perturber l’utilisation de votre téléphone. Elle épargnera le temps de devoir rechercher dans la zone de notification si l’alerte qui vient de produire est vraiment la confirmation d’une transaction ou bien autres choses.

**Badge de numérotation** : Il permet de notifier l’utilisateur en affichant sur son écran le nombre de transaction qu’il n’a pas encore vu. Nous avons fait en sorte qu’il soit de couleur bien tape-œil pour qu’on ne le rate pas.

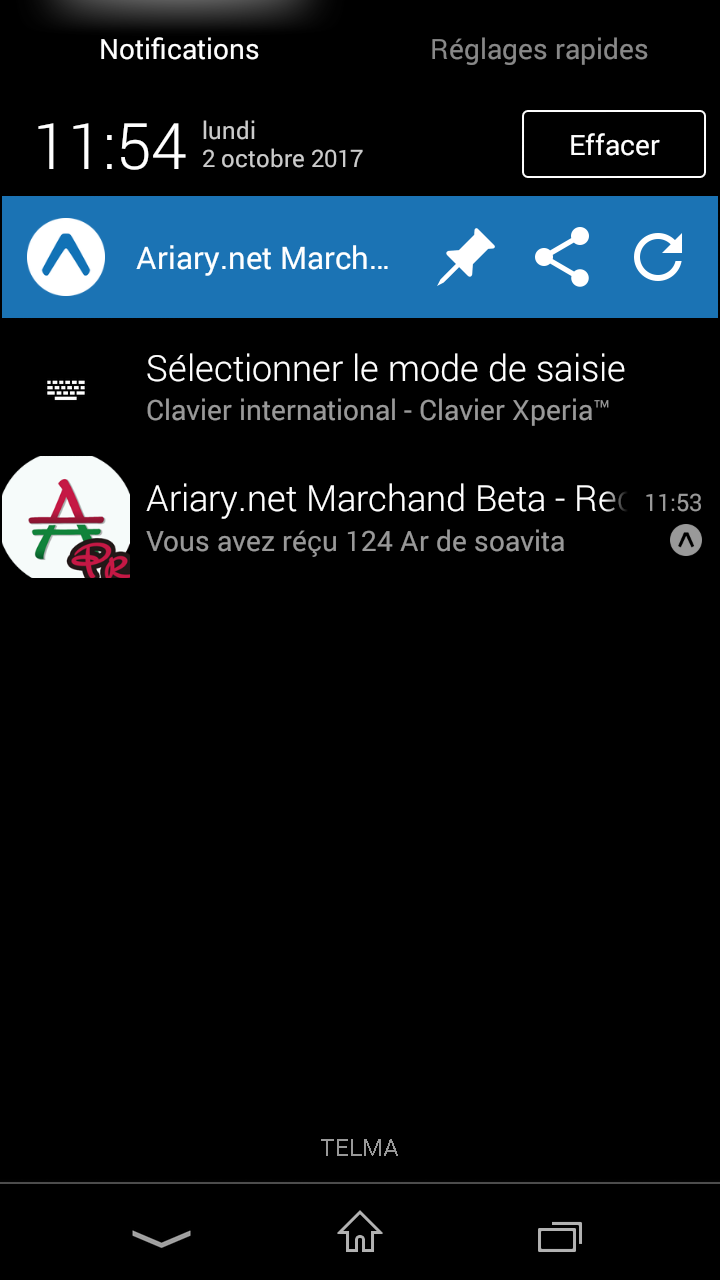
****

Figure 17 : notification en arrière-plan

Figure 18 : Badge de numérotation

Figure 19 : notification rapide

Notre système de notification marche grâce à la création d’un identifiant propre au compte dans le téléphone. Avec cet identifiant, on peut lui faire signe quand une transaction vient d’aboutir sans qu’il soit en écoute. On appelle ce système du « Push notification »

L’identification en question se crée à partir d’une librairie d’Expo permettant de générer une unique clé permettant d’envoyer une notification au mobile. La génération se base sur la combinaison de l’identifiant du téléphone et de l’application Ariary.net.

On aura donc un identifiant de ce genre :

ExponentPushToken[83D9aJCoDivkqZzPSKL7v8]

Cet identifiant est généré lors de la première utilisation et connexion à l’application. Elle sera ensuite envoyée dans le serveur pour la sauvegarde et son utilisation lors d’un envoi de notification.

Le système de notification suit les schémas suivants :

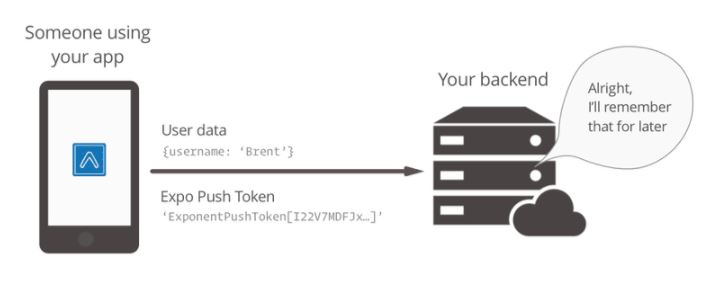


Figure 20 : génération de l'identifiant et sauvegarde dans le serveur

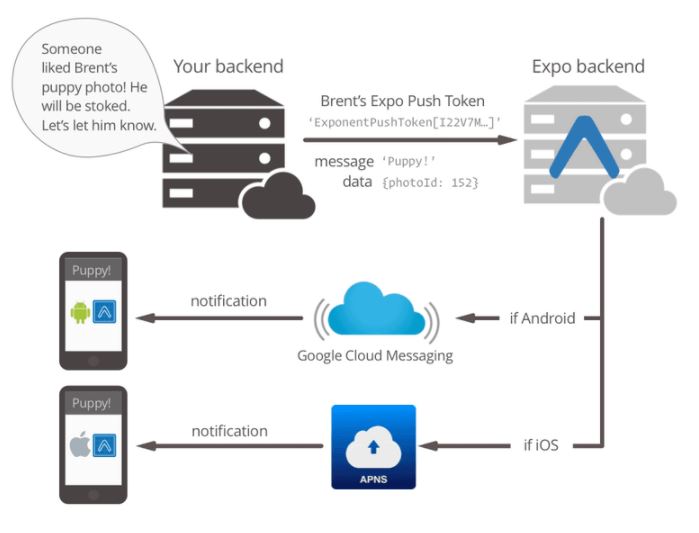


Figure 21 : Notification depuis le serveur

**Source :** https://docs.expo.io/versions/latest/guides/push-notifications.html

#### Historique de transaction et solde

Pour renforcer la fiabilité citée ci-dessus, un système de suivi des transactions et un affichage des soldes sont implémentés dans l’application.

On a fait en sorte que le solde du compte soit accessible à n’importe quelle partie de l’application. De ce fait, on l’a mis sur le menu de l’application permettant ainsi une vision rapide.

A chaque nouvelle notification, le solde se met à jour automatiquement mais le mode manuel est aussi disponible si besoin.

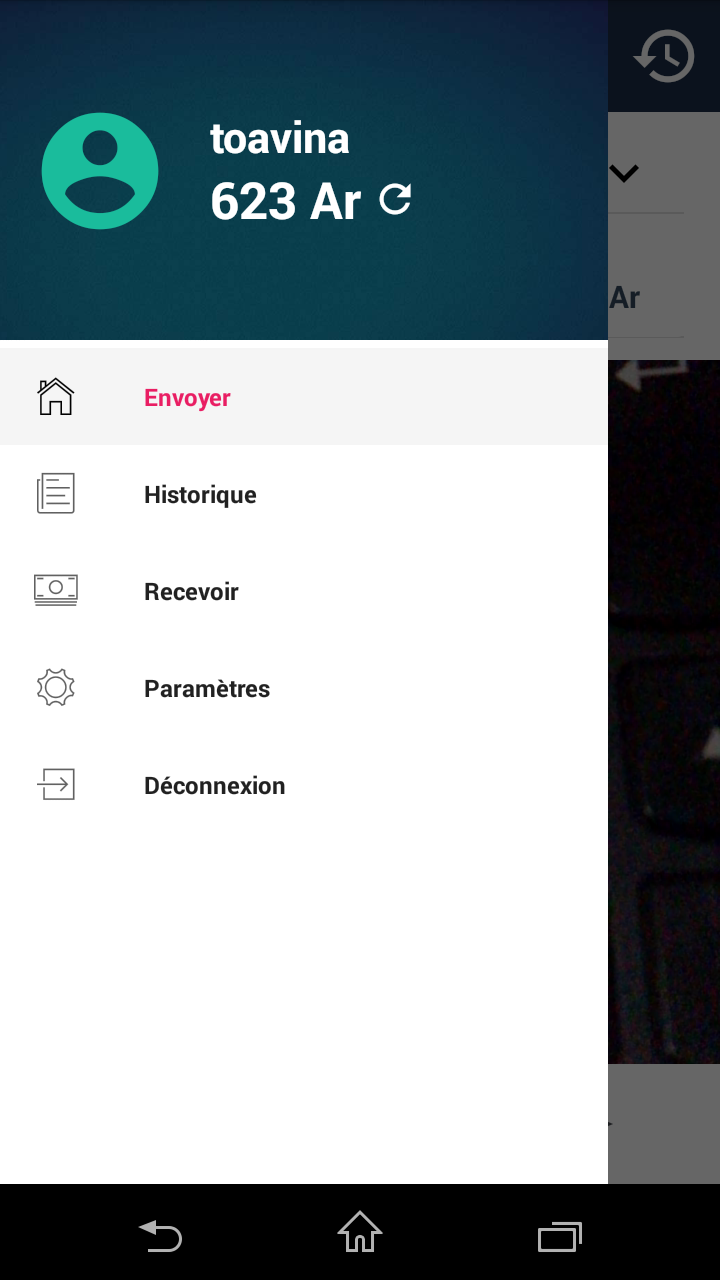


Figure 22 : Solde de compte et liste des menus

Dans un système d’échange monétaire, qu’est-ce qu’il y a de mieux qu’une liste de toutes les actions effectuées pour avoir la confiance d’un utilisateur. C’est dans cette mentalité que nous avons mise en place l’historique de toutes les transactions. Cette fonctionnalité permet un bon suivi des activités et on peut y avoir accès n’importe où dans l’application. (Sur la liste des menus et dans chaque page disponible).

Pour faciliter l’expérience utilisateur, nous avons fait des améliorations pour ne pas avoir une liste monotone :

* Regroupement de la liste par date (permettant ainsi de se situer très rapidement)
* Onglet regroupant les données par type de transaction (entrant ou sortant)
* Une recherche rapide par mot clé.

[Image d’illustration]

#### Système de sécurité

Hormis la question de fiabilité, on entend surtout parler de sécurité lorsqu’on rentre dans le domaine de la transaction monétaire. En suivant la norme du protocole de sécurité Oauth2, on a implémenté un système que l’on considère le plus sécurisé actuellement.

De ce fait, depuis la connexion d’un utilisateur jusqu’à chaque requête vers le serveur, nous ne sommes pas sortis de ce cadre et respecte chaque règle de sécurité.

Si nous parlons de la connexion d’utilisateur, le schéma des actions se déroule comme suit :

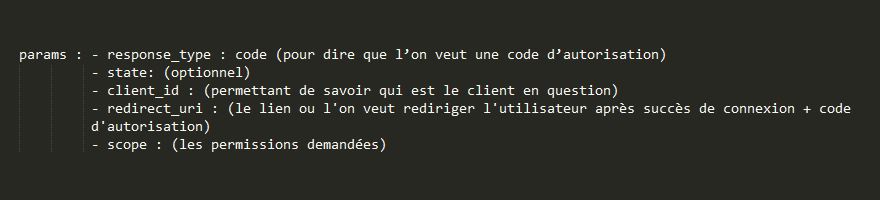
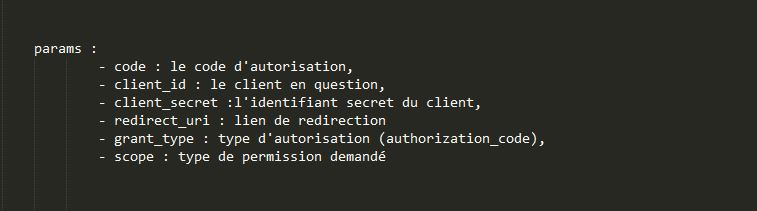
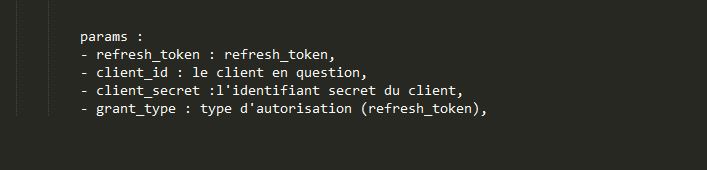
* + Dès qu’il veut se connecter, on redirige l’utilisateur vers la page de connexion d’Ariary.net.
  + Quand il a fini de se connecter, une « code d’autorisation » est générée et placée dans le lien (utilisable en une seule fois).

Figure 23 : Paramètres pour le code d'autorisation

* + Avec ce code d’autorisation, on peut ainsi demander un « Access Token » avec un « Refresh Token » permettant de faire les requêtes.
  + Et c’est enfin après que l’on peut avoir accès à des données avec l’Access Token (données de base de l’utilisateur, solde, historique…).
  + Le Refresh Token quant à lui sert à demander un nouvel Access Token si l’ancien n’est plus valide (utilisable une seule fois)

Voici une vision globale de l’utilisation de l’Oauth2 de type « Authorisation\_Code » sur un schéma :

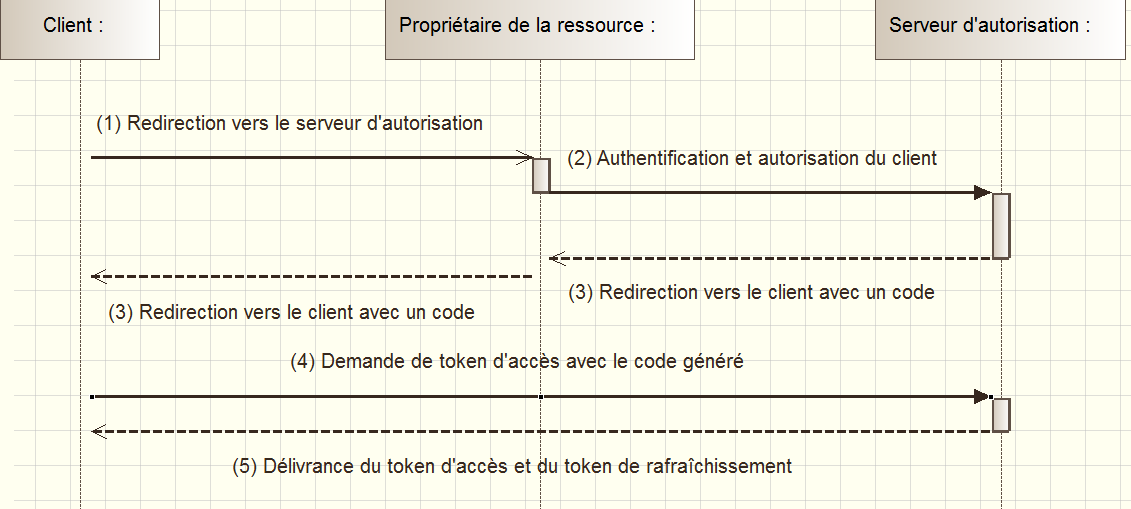


Figure 24 : Système Oauth2 - Type code d'autorisation

Source : <https://zestedesavoir.com/articles/1616/comprendre-oauth-2-0-par-lexemple/#2-les-roles>

#### Autres fonctionnalités

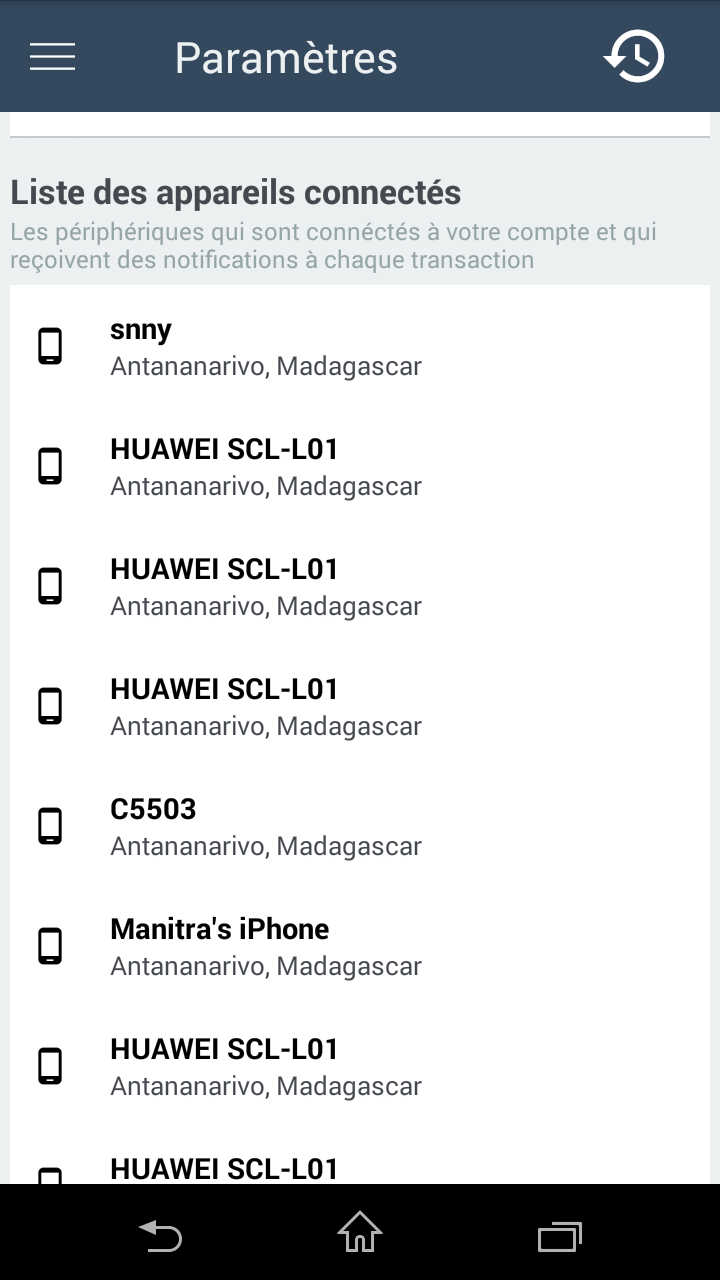
Toujours dans la question de sécurité, on peut trouver dans les paramètres la liste complète des appareils mobiles connectés à votre compte.

Figure 25 : Liste des appareils connectés

## État d’Analyse et Statistiques

### Volume de transaction Ariary.net

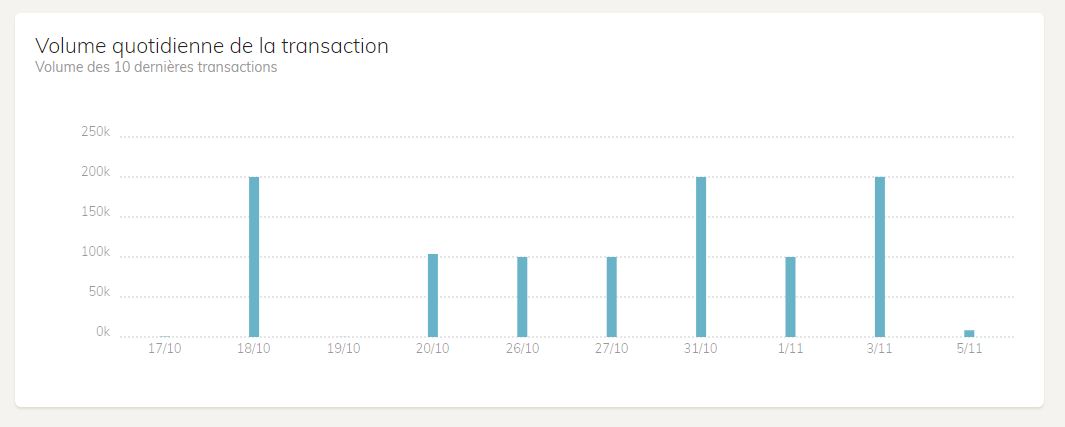
Pour le suivi des transactions propres à l’application mobile, la mise en place du serveur passerelle était d’une très grande utilité. Comme chaque transaction passe par ce serveur, on peut ériger des états propres à l’application et indépendant du serveur d’Ariary.net. On peut donc voir le volume quotidien et mensuel des transactions effectué par l’application mobile Ariary.net :

Figure 26 : Volume quotidien des transactions

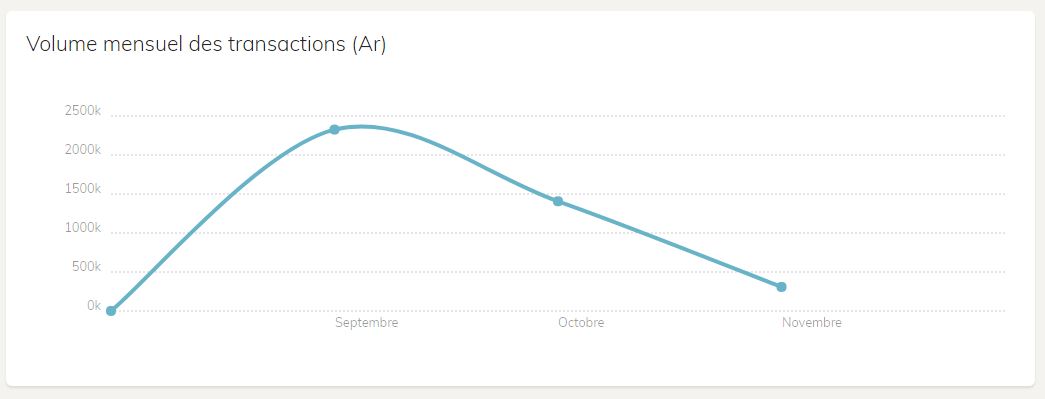


Figure 27 : Volume mensuel des transactions

## Problèmes rencontrés et solutions

Durant ce stage, nous avons vu divers problèmes et freins sur la création de l’application mobile, mais nous avons surtout quelque mal à faire un choix sur la technologie. A savoir entre le React-native et Native Scripts. Toutes les 2 sont des technologies que l’on n’a pas encore utilisées et bien nouvelles pour nous. Des recherches et des comparaisons ont été faites et notre choix s’est porté sur React-native par le fait qu’il offre déjà une libraire permettant de générer et de lire un QR Code, les fonctionnalités de fondamentale de notre application.

# Évaluation du projet et connaissances acquises

## Bilan pour l'entreprise

## Bilan personnel

Pour ma part, je peux considérer ce stage au sein de Nivo SA comme l’une des plus grandes expériences et la plus enrichissante que j’ai vécu jusqu’ici que ce soit du côté professionnel ou technique même de la vie en générale.

Pour le domaine technique, c’est à travers ce stage que j’ai appris énormément que ce soit en technologie ou méthodologie. La plupart de ce que l’on a utilisé m’était nouvelle et j’ai pu en tirer beaucoup de connaissance. Mais aussi le fait de travailler sur un produit que quelqu’un utilise vraiment donne l’envie d’en faire encore plus.

Sur le plan professionnel, le fait de voir et d’interagir avec de nouvelle personne m’a permis d’avoir de nouveaux contacts, de voir le monde d’une vision différentes que celui par la voie de l’informatique. Et surtout, leur partage a vraiment renforcé mon envie de ne pas rester indéfiniment travailler pour les autres mais de faire des choses pour mon propre compte.

## Extension et évolution de l’application

Notre application est actuellement en cours d’amélioration. Le but est d’offrir à l’utilisateur la meilleure expérience et nous avons déjà quelques idées pour cela :

* + Mise en place d’une autre application (basée sur Ariary.net) mais destinée principalement à des vendeurs ou des caisses de grandes surfaces. Le but est de ne permettre à cette application que la possibilité de recevoir pour la sécurité.
  + De gérer les mobiles connectés à un compte utilisateur depuis l’application lui-même.
  + D’y introduire les fonctions recharges de compte déjà développées ultérieurement.

# Conclusion

Durant ces quelques mois de stage chez Nivo SA, j’ai eu la chance d’appliquer dans la vraie vie et de rendre utile ce que j’avais acquis durant ces 3 ans d’étude.

Le module de taux de change Odoo a permis à l’entreprise d’avoir plus de précision dans ces actions. D’épargner une tache manuelle, certes de courte durée, qui pourrait comporter des erreurs d’insertion. Sauf un changement majeur sur le site de la Banque Centrale, le module « currency\_rate\_update\_MGA » peut continuer de marcher automatiquement dans Odoo.

L’application que nous avons développée nous a donné une vision et une preuve du potentiel que le système Ariary.net peut apporter. Donnant ainsi les forces et les motivations pour continuer à faire de notre mieux. Certes, elle n’est pas encore sortie officiellement mais dans l’idée de le faire très prochainement, nous travaillons toujours pour l’amélioration de ce que nous avons déjà.

Si on se réfère à ce que l’on s’est dit au début du projet, on peut affirmer que nous avons atteint les objectifs que nous nous sommes fixés, et on peut dire que l’on a même ajouté d’autres fonctionnalités en plus.

Ainsi, j’ai terminé mon stage de fin d’étude chez Nivo SA pour l’obtention du diplôme de Licence en Informatique dans l’option développement d’application. Durant cette période, j’ai eu la chance de passer du théorique au pratique et de travailler vraiment dans le monde professionnel. Et j’ai pu affiner les objectifs de ce que j’ai me suis fixé dans la vie.

# Bibliographie

Module Odoo de taux de change :

Odoo 9.0 : *Building a module :* Odoo documentation [*https://www.odoo.com/documentation/9.0/howtos/backend.html*](https://www.odoo.com/documentation/9.0/howtos/backend.html)

Docker : *Docker and Odoo :*

<https://docs.docker.com/samples/library/odoo/>

<https://hub.docker.com/_/odoo/>

BeautifulSoup : *Parser du HTML avec BeautifulSoup*

<http://sametmax.com/parser-du-html-avec-beautifulsoup/>

Application mobile Client/marchand :

React-native, *Official Documentation by Facebook.* Lien : <https://facebook.github.io/react-native/docs/getting-started.html>

Grafikart.fr, *Tutoriel JavaScript, Découverte de React-native, App météo [Formation sur YouTube].* Disponible sur : <https://www.youtube.com/watch?v=Y7rbJRjaYCY&t=369s>

Mohamad Mohebiafar, *React-native Tutorial* [*Formation sur YouTube].* Disponible sur : <https://www.youtube.com/watch?v=FUdy4PHIbww&t=19s>

React-native elements UI Toolkit, *Liste et exemple d’utilisation des composants de base de React-native. Lien :* [*https://react-native-training.github.io/react-native-elements*](https://react-native-training.github.io/react-native-elements)

Exponent API, *Official documentation by Expo.* Lien : <https://docs.expo.io/versions/latest/introduction/index.html>

Documentation sur le protocole Oauth2 :

Comprendre Oauth2, disponible sur :

<https://tools.ietf.org/html/rfc6749>

<https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>

<http://www.bubblecode.net/fr/2016/01/22/comprendre-oauth2/>

Dépôt de librairies JavaScript et React-native :

<https://www.npmjs.com/package/package>

<https://js.coach/react-native>

# Annexe

**Plugin Ariary.net pour WooCommerce**

Dans cette idée de lancement du système Ariary.net, on m’a aussi demandé de créer une extension WordPress pour le système. Ainsi, si on a un site WordPress avec WooCommerce (un plugin de commerce). On pour alors ajouter l’extension et de faire des ventes et recevoir des paiements directement dans son compte.

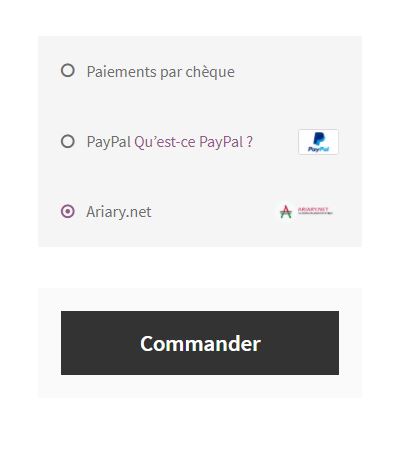
Cette extension se base sur la redirection vers le site d’Ariary.net si vous la choisissez comme moyen de paiement.

Figure 28 : liste des modes de paiement dans WordPress

**Classement des Bases de données nov. 2017 selon DB Engine Ranking**

Tableau 2 : Classement des bases de données

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | | | DBMS | Database Model | Score | | |
| Nov 2017 | **Oct 2017** | **Nov 2016** | **Nov 2017** | **Oct 2017** | **Nov 2016** |
| 1. | 1. | 1. | [Oracle detailed information](https://db-engines.com/en/system/Oracle) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 1360.05 | +11.25 | -52.96 |
| 2. | 2. | 2. | [MySQL detailed information](https://db-engines.com/en/system/MySQL) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 1322.03 | +23.20 | -51.53 |
| 3. | 3. | 3. | [Microsoft SQL Server detailed information](https://db-engines.com/en/system/Microsoft+SQL+Server) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 1215.08 | +4.76 | +1.27 |
| 4. | 4. | 4. | [PostgreSQL detailed information](https://db-engines.com/en/system/PostgreSQL) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 379.92 | +6.64 | +54.10 |
| 5. | 5. | 5. | [MongoDB detailed information](https://db-engines.com/en/system/MongoDB) | [Document store](https://db-engines.com/en/article/Document+Stores) | 330.47 | +1.07 | +5.00 |
| 6. | 6. | 6. | [DB2 detailed information](https://db-engines.com/en/system/DB2) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 194.06 | -0.53 | +12.61 |
| 7. | 7. | up arrow 8. | [Microsoft Access](https://db-engines.com/en/system/Microsoft+Access) | [Relational DBMS](https://db-engines.com/en/article/RDBMS) | 133.31 | +3.86 | +7.34 |
| 8. | 8. | down arrow 7. | [Cassandra detailed information](https://db-engines.com/en/system/Cassandra) | [Wide column store](https://db-engines.com/en/article/Wide+Column+Stores) | 124.21 | -0.58 | -9.76 |
| 9. | 9. | 9. | [Redis detailed information](https://db-engines.com/en/system/Redis) | [Key-value store](https://db-engines.com/en/article/Key-value+Stores) | 121.18 | -0.87 | +5.64 |
| 10. | 10. | up arrow 11. | [Elasticsearch detailed information](https://db-engines.com/en/system/Elasticsearch) | [Search engine](https://db-engines.com/en/article/Search+Engines) | 119.41 | -0.82 | +16.84 |

1. Source : https://fr.wikipedia.org/wiki/Standalone [↑](#footnote-ref-1)
2. TIC : Technologies de l’Information et de la Communication [↑](#footnote-ref-2)
3. Lien : http://ariary.net/ [↑](#footnote-ref-3)
4. Lien : http://www.banque-centrale.mg/ [↑](#footnote-ref-4)
5. Source : https://www.camptocamp.com/solution/odoo/ [↑](#footnote-ref-5)
6. Source : https://fr.wikipedia.org/wiki/Docker\_(logiciel) [↑](#footnote-ref-6)
7. Source : https://www.safaribooksonline.com/library/view/learning-react-native/9781491929049/ch01.html [↑](#footnote-ref-7)
8. Exemple : moment, axios, lodash, … [↑](#footnote-ref-8)
9. Expo XDE : [↑](#footnote-ref-9)
10. Banque centrale malgache : http://www.banque-centrale.mg/ [↑](#footnote-ref-10)
11. cURL : bibliothèque de requêtes aux URL pour les clients [↑](#footnote-ref-11)